Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application.

Claims 1-22. (Cancelled).

23. (Previously Presented) A superscalar microprocessor capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor comprising:

(a) an instruction fetch unit configured to provide a plurality of instructions to an instruction buffer;

(b) an execution unit, coupled to the instruction fetch unit, configured to execute the plurality of instructions from the instruction buffer in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of the program order so the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the program order, the load store unit including:

(i) an address path adapted to manage load and store addresses and to provide the load and store addresses to the memory system;

(ii) load dependency detection circuitry, wherein the load store unit does not make a particular load request when the load dependency detection circuitry detects an address collision or write pending for that particular load request; and

(iii) a data path adapted to transfer data from the memory system to the execution unit in response to load requests, the data path configured to align data returned from the memory system to thereby permit data falling on a word boundary to be returned from the memory system to the execution unit in correct alignment,

wherein the superscalar microprocessor initiates execution of more than one of the plurality of instructions from the instruction buffer in a clock cycle.

24. (Previously Presented) The microprocessor according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, wherein an address for a load request may be generated out-of-order.

25. (Previously Presented) The microprocessor according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, wherein an address for a store request may be generated out-of-order.

26. (Previously Presented) A superscalar microprocessor capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor comprising:

(a) an instruction fetch unit configured to provide a plurality of instructions to an instruction buffer;

(b) an execution unit, coupled to the instruction fetch unit, configured to execute the plurality of the instructions from the instruction buffer in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of the program order so that the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the program order, the load store unit having,

(i) an address generation unit configured to generate load and store addresses for instructions in the instruction buffer, wherein at least one of a load address and a store address may be generated out of the program order;

(ii) an address path adapted to manage the generated load and store addresses and to provide the generated load and store addresses to the memory system; and

(iii) a data path configured to transfer load data from the memory system to the execution unit,

wherein the superscalar microprocessor initiates execution of more than one of the plurality of instructions from the instruction buffer in a clock cycle.

27. (Previously Presented) The microprocessor according to claim 26, further including alignment control circuitry configured to generate a plurality of memory requests in response to a single instruction in the plurality of instructions when an operand of the single instruction falls on a word boundary.

28. (Previously Presented) The microprocessor according to claim 27, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

29. (Previously Presented) The microprocessor according to claim 27, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

30. (Previously Presented) The microprocessor according to claim 26, wherein the load store unit comprises dependency detection circuitry adapted to detect store-to-load dependencies, wherein the dependency detection circuitry determines when data for a load request depends on a store request.

31. (Previously Presented) The microprocessor according to claim 30, wherein the dependency detection circuitry includes address comparison logic configured to compare an address of a load request and an address of a store request.

32. (Previously Presented) The microprocessor according to claim 30, wherein the dependency detection circuitry includes relative age determining logic configured to determine the relative program order of a load request corresponding to a first memory instruction in the plurality of instructions and a store request corresponding to a second memory instruction in the plurality of instructions.

33. (Previously Presented) A computer system, comprising:

(a) a memory system configured to retain instructions and data, the instructions having a program order;

(b) a superscalar processor configured to execute the instructions, wherein the superscalar processor is configured to initiate more than one instruction in a clock cycle, the processor having,

(1) an instruction fetch unit configured to provide a plurality of instructions to an instruction buffer;

(2) an execution unit, coupled to the instruction fetch unit, configured to execute the plurality of instructions from the instruction buffer in an out-of-order fashion, the execution unit including,

(i) a register file;

(ii) address generation circuitry adapted to generate addresses for load requests and store requests out-of-order and

(iii) a load store unit adapted to make the load requests and the store requests to the memory system, the load store unit adapted to make at least one load request out of the program order so that the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the program order, the load store unit further adapted to return data falling on a word boundary in correct alignment to the register file.

34. (Previously Presented) The system according to claim 33, wherein the address generation circuitry is further adapted to generate addresses for the load and store requests as soon as all operands are valid and the address generation circuitry is available for address generation.

35. (Previously Presented) The system according to claim 33, wherein the generated addresses include linear and physical addresses, and the address generation circuitry is further adapted to general physical addresses corresponding to linear addresses.

36. (Previously Presented) The system according to claim 33, wherein the load store unit includes alignment control circuitry configured to generate a plurality of memory requests in response to a single instruction in the plurality of instructions when an operand of the single instruction falls on a word boundary.

37. (Previously Presented) The system according to claim 36, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

38. (Previously Presented) The system according to claim 36, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

39. (Previously Presented) The system according to claim 33, wherein the load store unit comprises dependency detection circuitry adapted to detect store-to-load dependencies, wherein the dependency detection circuitry determines when data for a load request depends on a store request.

40. (Previously Presented) The system according to claim 39, wherein the dependency detection circuitry includes address comparison logic configured to compare an address of a load request and an address of a store request.

41. (Previously Presented) The system according to claim 39, wherein the dependency detection circuitry includes relative age determining logic configured to determine the relative program order of a load request corresponding to a first memory instruction in the plurality of instructions and a store request corresponding to a second memory instruction in the plurality of instructions.

42. (Previously Presented) A superscalar microprocessor capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor comprising:

(a) an instruction fetch unit configured to provide a plurality of the instructions to an instruction buffer;

(b) an execution unit, coupled to the instruction fetch unit, configured to execute the plurality of instructions from the instruction buffer in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of the program order so that the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, and wherein the second instruction precedes the first instruction in the program order, the load store unit having,

(i) an address generation unit configured to generate load and store addresses for instructions in the instruction buffer, wherein at least one of a load address and a store address may be generated out of the program order;

(ii) an address path adapted to manage the generated load and store addresses and to provide the generated load and store addresses to the memory system;

(iii) dependency detection circuitry adapted to detect store-to-load dependencies, wherein the dependency detection circuitry determines when data for a load request depends on a store request; and

(iv) a data path configured to transfer load data from the memory system to the execution unit, the data path configured to align data returned from the memory system to thereby permit data falling on a word boundary to be returned from the memory system to the execution unit in correct alignment, wherein the superscalar microprocessor initiates execution of more than one of the plurality of instructions from the instruction buffer in a clock cycle.

43. (Previously Presented) The microprocessor according to claim 42, further including alignment control circuitry configured to generate a plurality of memory requests in response to a single instruction in the plurality of instructions when an operand of the single instruction falls on a word boundary.

44. (Previously Presented) The microprocessor according to claim 43, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

45. (Previously Presented) The microprocessor according to claim 43, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

46. (Previously Presented) The microprocessor according to claim 42, wherein the dependency detection circuitry includes relative age determining logic configured to determine the relative program order of a load instruction in the plurality of the instructions and a store instruction in the plurality of the instructions.

47. (Previously Presented) The microprocessor according to claim 23, wherein the load store unit is further adapted to make store requests in the program order.

48. (Previously Presented) The microprocessor according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests when all operands are valid and the address generation circuitry is available for address generation.

49. (Previously Presented) The microprocessor according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate linear addresses for the load and store requests, the linear address generation including the addition of three or more address components, the address components including a segment base, a base register, and a scaled index register.

50. (Previously Presented) The microprocessor according to claim 23, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, including generation of linear addresses and corresponding physical addresses.

51. (Previously Presented) The microprocessor according to claim 23, wherein the load store unit is further adapted to make memory-mapped input/output (I/O) requests according to the program order.

52. (Previously Presented) The microprocessor according to claim 23, wherein the data path is further adapted to merge data returning from the memory system with initial contents of a destination register.

53. (Previously Presented) The microprocessor according to claim 26, wherein the load store unit is further adapted to make store requests in the program order.

54. (Previously Presented) The microprocessor according to claim 26, wherein the address generation unit is further configured to generate load and store addresses when all operands are valid and the address generation unit is available for address generation.

55. (Previously Presented) The microprocessor according to claim 26, wherein the generated load and store addresses include linear and physical addresses, and the address generation unit is further configured to generate physical addresses corresponding to linear addresses.

56. (Previously Presented) The microprocessor according to claim 26, wherein the load store unit is adapted to make memory-mapped input/output (I/O) requests according to the program order.

57. (Previously Presented) The microprocessor according to claim 26, wherein the data path is further adapted to merge data returning from the memory system with initial contents of a destination register.

58. (Previously Presented) The system according to claim 33, wherein the load store unit is further adapted to make store requests in the program order.

59. (Previously Presented) The system according to claim 33, wherein the load store unit is further adapted to make memory-mapped input/output (I/O) load requests in the program order.

60. (Previously Presented) The system according to claim 33, wherein the load store unit is further adapted to merge data returning from the memory system with initial contents of a destination register.

61. (Previously Presented) The microprocessor according to claim 42, wherein the load store unit is further adapted to make store requests in the program order.

62. (Previously Presented) The microprocessor according to claim 42, wherein the address generation unit is further configured to generate load and store addresses as soon as all operands are valid and the address generation unit is available for address generation.

63. (Previously Presented) The microprocessor according to claim 42, wherein the address generation unit is further configured to generate linear load and store addresses, the linear address generation including the addition of three or more address components, the address components including a segment base, a base register, and a scaled index register.

64. (Previously Presented) The microprocessor according to claim 42, wherein the address generation unit is further configured to generate linear load and store addresses, the linear address generation including the addition of three or more address components, the address components including a segment base, a base register, and a displacement.

65. (Previously Presented) The microprocessor according to claim 42, wherein the generated load and store addresses include linear and physical addresses, and the address generation unit is further configured to generate physical addresses corresponding to linear addresses.

66. (Previously Presented) The microprocessor according to claim 42, wherein the load store unit is further adapted to make memory-mapped input/output (I/O) load requests in the program order.

67. (Previously Presented) The microprocessor according to claim 42, wherein the data path is further adapted to merge data returning from the memory system with initial contents of a destination register.

68. (Previously Presented) A superscalar microprocessor capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor comprising:

an execution unit configured to execute a plurality of instructions in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of the program order so the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the program order, the load store unit including:

(i) an address path adapted to manage load and store addresses and to provide the load and store addresses to the memory system;

(ii) load dependency detection circuitry, wherein the load store unit does not make a particular load request when the load dependency detection circuitry detects an address collision or write pending for that particular load request; and

(iii) a data path adapted to transfer data from the memory system to the execution unit in response to load requests, the data path configured to align data returned from the memory system to thereby permit data falling on a word boundary to be returned from the memory system to the execution unit in correct alignment,

wherein the superscalar microprocessor initiates execution of more than one of the plurality of instructions in a clock cycle.

69. (Previously Presented) The microprocessor according to claim 68, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, wherein an address for a load request may be generated out-of-order.

70. (Previously Presented) The microprocessor according to claim 68, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, wherein an address for a store request may be generated out-of-order.

71. (Previously Presented) The microprocessor according to claim 68, wherein the load store unit is further adapted to make store requests in the program order.

72. (Previously Presented) The microprocessor according to claim 68, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests when all operands are valid and the address generation circuitry is available for address generation.

73. (Previously Presented) The microprocessor according to claim 68, wherein the execution unit further comprises address generation circuitry adapted to generate linear addresses for the load and store requests, the linear address generation including the addition of three or more address components, the address components including a segment base, a base register, and a scaled index register.

74. (Previously Presented) The microprocessor according to claim 68, wherein the execution unit further comprises address generation circuitry adapted to generate addresses for the load and store requests, including generation of linear addresses and corresponding physical addresses.

75. (Previously Presented) The microprocessor according to claim 68, wherein the load store unit is further adapted to make memory-mapped input/output (I/O) load requests according to the program order.

76. (Previously Presented) The microprocessor according to claim 68, wherein the data path is further adapted to merge data returning from the memory system with initial contents of a destination register.

77. (Previously Presented) The microprocessor according to claim 68, wherein the execution unit is further configured to merge data returning from the memory system with initial contents of a destination register.

78. (Previously Presented) The microprocessor according to claim 68, further comprising an instruction fetch unit configured to provide the plurality of instructions to an instruction buffer, wherein the execution unit executes the plurality of instructions from the instruction buffer in an out of order fashion.

79. (Previously Presented) A superscalar microprocessor capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor comprising:

an execution unit configured to execute a plurality of instructions in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of the program order so that the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the program order, the load store unit having,

(i) an address generation unit configured to generate load and store addresses out of order for instructions in the plurality of instructions;

(ii) an address path adapted to manage the generated load and store addresses and to provide the generated load and store addresses to the memory system; and

(iii) a data path configured to transfer load data from the memory system to the execution unit, wherein the superscalar microprocessor initiates execution of more than one of the plurality of instructions in a clock cycle.

80. (Previously Presented) The microprocessor according to claim 79, further including alignment control circuitry configured to generate a plurality of memory requests in response to a single instruction in the plurality of instructions when an operand of the single instruction falls on a word boundary.

81. (Previously Presented) The microprocessor according to claim 80, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

82. (Previously Presented) The microprocessor according to claim 80, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

83. (Previously Presented) The microprocessor according to claim 79, wherein the load store unit comprises dependency detection circuitry adapted to detect store-to-load dependencies, wherein the dependency detection circuitry determines when data for a load request depends on a store request.

84. (Previously Presented) The microprocessor according to claim 83, wherein the dependency detection circuitry includes address comparison logic configured to compare an address of a load request and an address of a store request.

85. (Previously Presented) The microprocessor according to claim 83, wherein the dependency detection circuitry includes relative age determining logic configured to determine the relative program order of a load request corresponding to a first memory instruction in the plurality of instructions and a store request corresponding to a second memory instruction in the plurality of instructions.

86. (Previously Presented) The microprocessor according to claim 79, wherein the load store unit is further adapted to make store requests in the program order.

87. (Previously Presented) The microprocessor according to claim 79, wherein the address generation unit is further configured to generate load and store addresses when all operands are valid and the address generation unit is available for address generation.

88. (Previously Presented) The microprocessor according to claim 79, wherein the generated load and store addresses include linear and physical addresses, and the address generation unit is further configured to generate physical addresses corresponding to linear addresses.

89. (Previously Presented) The microprocessor according to claim 79, wherein the load store unit is adapted to make memory-mapped input/output (I/O) load requests according to the program order.

90. (Previously Presented) The microprocessor according to claim 79, wherein the execution unit is further configured to merge data returning from the memory system with initial contents of a destination register.

91. (Previously Presented) The microprocessor according to claim 79, wherein the data path is further configured to merge data returning from the memory system with initial contents of a destination register.

92. (Previously Presented) The microprocessor according to claim 79, further comprising an instruction fetch unit configured to provide the plurality of instructions to an instruction buffer, wherein the execution unit executes the plurality of instructions from the instruction buffer in an out of order fashion.

93. (Currently amended) A superscalar microprocessor configured to initiate execution of more than one instruction in a clock cycle, the processor comprising:

(a) a memory system configured to retain instructions and data, the instructions having a program order;

(b) an execution unit configured to execute the plurality of instructions in an out-of-order fashion, the execution unit including,

(~~iii~~ i) a register file;

(~~iv~~ ii) address generation circuitry adapted to generate addresses for load requests and store requests out-of-order; and

(iii) a load store unit adapted to make the load requests and the store requests to the memory system, the load store unit adapted to make at least one load request out of the program order so that the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the program order, the load store unit further adapted to return data falling on a word boundary in correct alignment to the register file.

94. (Previously Presented) The microprocessor according to claim 93, wherein the address generation circuitry is further adapted to generate addresses for the load and store requests when all operands are valid and the address generation circuitry is available for address generation.

95. (Previously Presented) The microprocessor according to claim 93, wherein the generated addresses include linear and physical addresses, and the address circuitry is further adapted to generate physical addresses corresponding to linear addresses.

96. (Previously Presented) The microprocessor according to claim 93, wherein the load store unit includes alignment control circuitry configured to generate a plurality

of memory requests in response to a single instruction in the plurality of instructions when an operand of the single instruction falls on a word boundary.

97. (Previously Presented) The microprocessor according to claim 96, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

98. (Previously Presented) The microprocessor according to claim 96, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

99. (Previously Presented) The microprocessor according to claim 93, wherein the load store unit comprises dependency detection circuitry adapted to detect store-to-load dependencies, wherein the dependency detection circuitry determines when data for a load request depends on a store request.

100. (Previously Presented) The microprocessor according to claim 99, wherein the dependency detection circuitry includes address comparison logic configured to compare an address of a load request and an address of a store request.

101. (Previously Presented) The microprocessor according to claim 99, wherein the dependency detection circuitry includes relative age determining logic configured to determine the relative program order of a load request corresponding to a

first memory instruction in the plurality of instructions and a store request corresponding to a second memory instruction in the plurality of instructions.

102. (Previously Presented) The microprocessor according to claim 93, wherein the load store unit is further adapted to make store requests in the program order.

103. (Previously Presented) The microprocessor according to claim 93, wherein the load store unit is further adapted to make memory-mapped input output (I/O) load requests in the program order.

104. (Previously Presented) The microprocessor according to claim 93, wherein the load store unit is further adapted to merge data returning from the memory system with initial contents of a destination register.

105. (Previously Presented) The microprocessor according to claim 93, wherein the execution unit further includes merge data circuitry configured to merge data returning from the memory system with initial contents of a destination register.

106. (Previously Presented) The microprocessor according to claim 93, further comprising an instruction fetch unit configured to provide the plurality of instructions to an instruction buffer, wherein the execution unit executes the plurality of instructions from the instruction buffer in an out of order fashion.

107. (Previously Presented) A superscalar microprocessor capable of executing one or more instructions out-of-order with respect to an ordering defined by a program order, the microprocessor comprising:

an execution unit configured to execute a plurality of instructions in an out-of-order fashion, the execution unit including a load store unit adapted to make load requests and store requests to a memory system, the load store unit adapted to make at least one load request out of the program order so that the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, and wherein the second instruction precedes the first instruction in the program order, the load store unit having,

(i) an address generation unit configured to generate load and store addresses out of order for instructions in the plurality of instructions;

(ii) an address path adapted to manage the generated load and store addresses and to provide the generated load and store addresses to the memory system;

(iii) dependency detection circuitry adapted to detect store-to-load dependencies, wherein the dependency detection circuitry determines when data for a load request depends on a store request; and

(iv) a data path configured to transfer load data from the memory system to the execution unit, the data path configured to align data returned from the memory system to thereby permit data falling on a word boundary to be returned from the memory system to the execution unit in correct alignment, wherein the superscalar microprocessor initiates execution of more than one of the plurality of instructions in a clock cycle.

108. (Previously Presented) The microprocessor according to claim 107, further including alignment control circuitry configured to generate a plurality of memory requests in response to a single instruction in the plurality of instructions when an operand of the single instruction falls on a word boundary.

109. (Previously Presented) The microprocessor according to claim 108, wherein the single instruction is a load instruction and the plurality of memory requests are load requests.

110. (Previously Presented) The microprocessor according to claim 108, wherein the single instruction is a store instruction and the plurality of memory requests are store requests.

111. (Previously Presented) The microprocessor according to claim 107, wherein the dependency detection circuitry includes relative age determining logic configured to determine the relative program order of a load instruction in the plurality of the instructions and a store instruction in the plurality of the instructions.

112. (Previously Presented) The microprocessor according to claim 107, wherein the load store unit is further adapted to make store requests in the program order.

113. (Previously Presented) The microprocessor according to claim 107, wherein the address generation unit is further configured to generate load and store addresses as soon as all operands are valid and the address generation unit is available for address generation.

114. (Previously Presented) The microprocessor according to claim 107, wherein the address generation unit is further configured to generate linear load and store addresses, the linear address generation including the addition of three or more address components, the address components including a segment base, a base register, and a scaled index register.

115. (Previously Presented) The microprocessor according to claim 107, wherein the generated load and store addresses include linear and physical addresses, and the address generation unit is further configured to generate physical addresses corresponding to linear addresses.

116. (Previously Presented) The microprocessor according to claim 107, wherein the load store unit is further adapted to make memory-mapped input/output (I/O) load requests in the program order.

117. (Previously Presented) The microprocessor according to claim 107, wherein the data path is further configured to merge data returning from the memory system with initial contents of a destination register.

118. (Previously Presented) The microprocessor according to claim 107, wherein the load store unit includes merge data circuitry configured to merge data returning from the memory system with initial contents of a destination register.

119. (Previously Presented) The microprocessor according to claim 107, wherein the execution unit is further configured to merge data returning from the memory system with initial contents of a destination register.

120. (Previously Presented) The microprocessor according to claim 107, wherein the execution unit is further configured to provide store data to the data path as load data when the dependency detection circuitry determines that data for a load request depends on a store request.

121. (Previously Presented) The microprocessor according to claim 120, wherein the execution unit is further configured to provide data stored by a store request as load data by way of the memory system.

122. (Previously Presented) The microprocessor according to claim 107, further comprising an instruction fetch unit configured to provide the plurality of instructions to an instruction buffer, wherein the execution unit executes the plurality of instructions from the instruction buffer in an out of order fashion.

123. (New) In a superscalar microprocessor having an execution unit adapted to execute a plurality of instructions and to issue load instructions out-of-order, a method for managing requests for loads and stores to and from a memory device, the method comprising:

calculating an address for an instruction and transferring said address to a load store unit;

determining whether said instruction involves at least one of a load operation and a store operation;

checking, if said instruction has a load operation, for an address collision and for any write pendings, and signaling the outcome of said check;

making a request to said memory device based on a priority scheme and the results of said checking step, wherein said priority scheme includes making at least one load request out of an ordering so the one load request can be made before a memory request, wherein the one load request corresponds to a first instruction from the plurality of instructions and the memory request corresponds to a second instruction from the plurality of instructions, wherein the second instruction precedes the first instruction in the ordering;

receiving requested data from said load operation and/or said store operation in a data path portion of said load store unit; and

aligning said requested data if said requested data is unaligned.

124. (New) The method of claim 123, further comprising performing a data dependency check on said instruction prior to said address calculation.

125. (New) The method of claim 123, further comprising writing the results of said instruction into a preassigned location in a temporary buffer.

126. (New) The method of claim 125, further comprising providing data to said load store unit by bypassing said temporary buffer.

127. (New) The method of claim 123, further comprising preventing load bypassing of load operations that would otherwise incorrectly modify state of a system coupled to the microprocessor.

128. (New) The method of claim 123, further comprising merging data received from memory with data stored in a destination register.

129. (New) The method of claim 123, wherein said step of checking includes comparing the first address of said load operation against the first and last address for an older unretired store operation.

130. (New) A method for executing one or more instructions out of order using a superscalar microprocessor, the method comprising:

receiving a plurality of instructions having an ordering, the plurality of instructions including a store instruction and a load instruction;

generating an address for an instruction in the plurality of instructions, wherein the generated address includes an address that is generated out of order with respect to the ordering;

if the instruction involves a load operation, determining whether the load operation can be executed out of order;

if the load operation can be executed out of order, executing the load operation out of order using the generated address, including performing a load request corresponding to the load instruction before a store request corresponding to the store instruction, wherein the store instruction precedes the load instruction in the ordering;

receiving requested data from the load operation; and

aligning the requested data if said generated address is unaligned.

131. (New) The method of claim 130, wherein the determining whether the load operation can be executed out of order comprises comparing the generated address to a store address for the store request.

132. (New) The method of claim 131, wherein the determining whether the load operation can be executed out of order further comprises determining if the load operation depends on the memory request based on the comparison.

133. (New) The method of claim 132, wherein the determining whether the load operation can be executed out of order further comprises determining if out of order execution of the load operation would incorrectly modify a state of a system coupled to the microprocessor.

134. (New) The method of claim 133, further comprising merging the aligned data with initial data in a load destination register.

135. (New) The method of claim 134, further comprising writing results of the plurality of instructions into preassigned locations in a register file.

136. (New) A method for executing one or more instructions out of order using a superscalar microprocessor, the method comprising:

receiving a plurality of instructions having an ordering, the plurality of instructions including a store instruction and a load instruction, the store instruction being before the load instruction in the ordering;

generating a load address for the load instruction and a store address for the store instruction, wherein at least one of the load address and the store address is generated out of order with respect to the ordering;

comparing the load address to the store address;

determining, in part from the comparison, if the load instruction depends on the store instruction;

if the load instruction does not depend on the store instruction, then retiring at least a portion of data provided from a data cache according to the load address, the provided data having been aligned if the load address is unaligned; and

if the load instruction does depend on the store instruction, then retiring at least a portion of load data according to store data received for the store instruction.

137. (New) The method of claim 136, further comprising:

merging the at least a portion of data provided from the data cache with initial data from a load destination register; and

merging the at least a portion of load data according to store data with initial data from a load destination register.

138. (New) The method of claim 136, further comprising:

writing results of the plurality of instructions into preassigned locations in a register file;

storing at least one of the load address and the store address into a first one of a plurality of address buffers; and

wherein the comparing the load address to the store address comprises receiving contents of the first address buffer.

139. (New) The method of claim 136, further comprising preventing load bypassing of load operations that would otherwise incorrectly modify state of a system coupled to the microprocessor.

140. (New) The method of claim 136, wherein the comparing the load address to the store address includes determining if any byte referenced by the load instruction overlaps with any byte referenced by the store instruction.